Computational cost and accuracy in calculating three-dimensional radiative transfer: Results for new implementations of Monte Carlo and SHDOM

Robert Pincus [*]

University of Colorado and NOAA Earth System Research Laboratory

Boulder, CO, USA

K. Franklin Evans

Department of Atmospheric and Oceanic Sciences

University of Colorado, Boulder, CO, USA

[*] *Corresponding Author Address:* Robert Pincus, Cooperative Institute for Research in Environmental Sciences, University of Colorado/NOAA Earth Systems Research Laboratory Physical Sciences Division, 325 Broadway, R/PSD1, Boulder, CO, 80305. email: Robert.Pincus@colorado.edu.

ABSTRACT

This paper examines the trade-offs between computational cost and accuracy for two new state-of-the-art codes for computing three-dimensional radiative transfer: a community Monte Carlo model and a parallel implementation of the Spherical Harmonics Discrete Ordinates Method (SHDOM). Both codes are described and algorithmic choices are elaborated. Two prototype problems are considered: a domain filled with stratocumulus clouds and another containing scattered shallow cumulus, absorbing aerosols, and molecular scatterers. Calculations are performed for a range of resolutions and the relationships between accuracy and computational cost, measured by time-to-solution and memory use, are compared.

Monte Carlo accuracy depends primarily on the number of trajectories used in the integration. Monte Carlo estimates of intensity are computationally expensive and may be subject to large sampling noise from highly-peaked phase functions. This noise can be decreased using a range of variance reduction techniques but these techniques can compromise the excellent agreement between the true error and estimates obtained from unbiased calculations. SHDOM accuracy is controlled by both spatial and angular resolution; different output fields are sensitive to different aspects of this resolution, so that the optimum accuracy parameters depend on which quantities are desired as well as on the characteristics of the problem being solved. The accuracy of SHDOM must be assessed through convergence tests and all results from un-converged solutions may be biased.

SHDOM is more efficient (i.e. has lower error for a given computational cost) than Monte Carlo when computing pixel-by-pixel upwelling fluxes in the cumulus scene, while Monte Carlo is more efficient in computing flux divergence and downwelling flux in the stratocumulus scene, especially at higher accuracies. The two models are comparable for downwelling flux and flux divergence in cumulus and upwelling flux in stratocumulus. SHDOM is substantially more efficient when computing pixel-by-pixel intensity in multiple directions; the models are comparable when computing domain-average intensities. In some cases memory use, rather than computation time, may limit the resolution of SHDOM calculations.

1. Computational methods for solving the radiative transfer equation in three spatial dimensions

In the fundamental equation describing the transfer of monchromatic, unpolarized radiation in the atmosphere the radiation field depends on two angular dimensions (the zenith and azimuthal angles) and three spatial dimensions. In most applications of radiative transfer in the atmospheric sciences this equation is simplified by omitting the two horizontal spatial dimensions so that the radiation field depends only on angle and vertical location. The so-called ”1D” radiative transfer equation may be used repeatedly to account for variability in the horizontal dimension by assuming that there is not net transfer of radiation in the horizontal. But this assumption is not always valid, and differences between radiation fields computed using 1D and 3D radiative transfer can impact fluxes at the top and bottom of the atmosphere (see, e.g, Scheirer and Macke 2003), local heating rates (e.g. Di Giuseppe and Tompkins 2003), and especially the estimation of atmospheric properties from remotely-sensed radiation (e.g. Loeb et al. 1998; Marshak et al. 2006, among many others). Differences between 3D and 1D results are most acute for problems involving clouds at visible wavelengths, where multiple scattering is dominant and cloud extinction varies on spatial scales smaller than the radiation mean free path.

There are two main barriers to the use of three-dimensional radiative transfer. First, specifying the problem can be quite difficult, since this requires describing the instantaneous distribution of extinction, single-scattering albedo, and scattering phase function in three spatial dimensions at very small scales. No current observing system is up to this task, although fine-scale cloud models can provide useful test cases. Secondly, solving the three-dimensional radiative transfer equation is much more computationally expensive than solving its one-dimensional counterpart.

There are two broad classes of methods for computing radiative transfer in three-dimensionally inhomogeneous atmospheres (see, e.g., Evans and Marshak 2005). Explicit methods discretize the radiation field in space, using discrete grids or finite elements, and in angle, using discrete ordinates or spherical harmonics. The methods start with a first guess solution and iteratively adjust the radiation field until it agrees with the radiative transfer equation to some specified accuracy. The most widely-used explicit method in the atmospheric sciences is the spherical harmonics discrete ordinates method (SHDOM; Evans 1998).

Explicit methods compute the full spatially- and angularly-varying radiation field at once. This means that any and all output quantities, from domain-averaged top-of-atmosphere fluxes to cell-by-cell flux divergences, can be derived from the same basic calculation without much additional computation, but it also means that one must do a complete calculation even if only a subset of quantities are desired. The only way to assess accuracy when using explicit methods is to check for convergence as a function of algorithm accuracy parameters. All quantities from an un-converged solution, including domain averages, may be biased. Explicit methods are difficult to formulate as parallel algorithms, and are often limited by the amount of memory available for a calculation.

Monte Carlo methods, in contrast, solve the radiative transfer equation (expressed as an integral over orders of scattering) by computing a large number of discrete trajectories through the full radiation field. In the atmospheric sciences Monte Carlo methods are usually described as simulating the trajectories of individual photons, although this explanation is fundmentally incorrect (see, for example, Mishchenko 2008). Samples are followed from their introduction into the medium through a series of scattering and absorption events until they are either completely absorbed or exit the medium.

Monte Carlo methods are extremely flexible, allowing for the calculation of almost any quantity including some, like the distribution of path lengths, that can not be computed by explicit methods. Basic radiation transport algorithms allow for the computation of fluxes and flux divergences rates; intensity calculations add significant computational cost. Accuracy comes slowly in Monte Carlo calculations, increasing as the inverse square root of the number of samples, but the uncertainty can be estimated by dividing the total number of samples into batches and computing the standard error of the set of batches. Each trajectory represents an independent sample of the full solution, so integral quantities are more accurate than spatially- or angularly-resolved estimates for a given number of trajectories. Monte Carlo methods are embarrassingly parallel since any subset of the complete calculation, down to the level of individual trajectories, can be computed independently.

Both explicit and Monte Carlo methods can provide the correct answer to a given problem in three-dimensional radiative transfer given sufficient resources. The computational costs of the solution, however, can vary tremendously between the two approaches. The relative efficiency of the methods depends on what quantities are required and the size and complexity of the problem domain.

This paper examines the trade-offs between cost and accuracy for SHDOM, an explicit method recently reformulated as a parallel algorithm, and a freely-available Monte Carlo radiative transfer for monochromatic problems in unpolarized solar radiative transfer. The next section describes the new Monte Carlo model including details about the computational structure and algorithms; the following section describes how SHDOM has been adapted to run on multiple processors and how this impacts the computational cost. We describe two prototype problems in three-dimensional radiative transfer, consider the impact of resolution and algorithmic parameters in both models, then compare the relationships between computational cost and accuracy in solving these problems for both SHDOM and the Monte Carlo model. The final section distills these results and other considerations into guidelines for choosing the most computationally efficient method for a given radiative transfer problem.

Throughout the paper we assume that readers are generally familiar with both the SHDOM algorithm (Evans 1998) and Monte Carlo simulation of radiative transfer. For an overview of the latter subject we recommend Iwabuchi's (2006) lucid introduction or Evans and Marshak (2005) for more mathematical detail and rigor.

2. The I3RC Community Monte Carlo model

One of the goals of the International Intercomparison of 3-Dimensional Radiation Codes project (I3RC; see Cahalan et al. 2005) is to foster the development and sharing of tools for computing three-dimensional radiative transfer. At the time of the first I3RC workshop (November 1999) SHDOM was the only freely-distributed tool available for computing three-dimensional radiative transfer. Scientists wishing to use Monte Carlo methods had to write their own code or obtain one from a colleague. To facilitate the use of Monte Carlo methods the project sponsored the development of a Monte Carlo model, intended for use in both research and education, based on requirements developed by members of the I3RC community. We implemented this model, which has been available from `http://i3rc.gsfc.nasa.gov/I3RC_community_model.htm` since August 2006 and is also available on Google Code [`http://code.google.com/p/i3rc-monte-carlo-model/`]. Calculations in this paper were done with the ”Cornish-Gilliflower” release of March 2009. Spurred in part by results from the I3RC project, several other Monte Carlo codes have become available in the the intervening years including the MC-UNIK and Grimaldi models [`http://www.ifm-geomar.de/index.php?id=981`] and MCARATS [`http://www.geocities.jp/null2unity/mcarats/`].

a. Computational architecture

The I3RC Community Monte Carlo radiative transfer code is written as a series of modules and several programs in object-oriented Fortran 95. Each module corresponds to a different conceptual aspect of the problem (a single scattering phase function, an incoming stream of radiation, a domain filled with clouds, aerosols, and gases, and so on) and contains the definitions for one or more derived types and a set of procedures that operate on those derived types. Procedures outside a module do not have access to the internal data structures; this data must be accessed or modified through procedures defined by the module.

Modules may build on one another: the module used to describe the distribution of optical properties within a domain, for example, uses the variables describing scattering phase functions. The majority of the modules and the relationships between them are shown in Figure 1 .

Object-oriented programming is the practice of packaging data structures together with the procedures that act on them. The technique is not often used in the atmospheric sciences but has proved very valuable in other communities because it provides a way to divide a problem into conceptually separate parts. This makes it easier to test individual software elements as isolated units, which in turn helps programmers identify errors more quickly. Because objects store data locally it is possible to describe, validate, or solve a problem in steps, rather than assembling every piece of input data, output space, and algorithmic detail into a single subroutine call. Local copies of data also facilitate the use of code in parallel environments. Finally, object-oriented programming makes it possible to reuse elements of the software – one could write an interface that would apply a plane-parallel radiative transfer model to each column in a three-dimensionally variable domain, for example, or replace any module in our code with one that implemented the same procedures. There are two major drawbacks, both related to the encapsulation of data within objects: first, that extra coding is required to provide access to this data, and secondly that the code may be less efficient, in terms of processing time or memory, at completing tasks than architectures where data is manipulated directly.

The heart of the I3RC Community Monte Carlo model is a monochromatic integrator. This solver processes a series of samples (”photons”) using forward Monte Carlo to provide estimates of radiative quantities. The problem is specified by using other modules to describe the distribution of optical properties of an arbitrary number of components (e.g. clouds, aerosols, gases, etc.) within a three-dimensional domain. Scattering phase functions for each component are tabulated at discrete values of some parameter (typically the particle size) and a single entry from this table represents the phase function within each grid cell.

b. Algorithmic aspects

The integrator provided with the I3RC model implements both straightforward ray tracing and maximum cross-section (Marchuk et al. 1980) algorithms to compute the sample trajectories. We have found ray tracing to be more efficient in most applications. Cost increases for both algorithms with the optical depth of the medium and the number of cells into which the medium is divided. Panel a of Figure 2 shows this dependency for a conservatively-scattering plane-parallel medium with scattering phase functions appropriate to a distribution of cloud drops with effective radius 8.8 μm, normalized by the time for the simplest case (lowest optical depth and single-celled domain).

The integrator can also provide estimates of intensity in arbitrary directions using local estimation. This technique involves accumulating the extinction from each scattering location to the boundary of the medium along each direction in which intensity is desired, so it requires additional computational time that increases with the number of cells that must be traced to the boundary at each scattering event. The solid lines in panel b of Fig. 2 quantify this cost in a plane-parallel medium.

c. Methods for reducing computation time

Most Monte Carlo models treat absorption by mutiplying the weight w of each sample by the single scattering albedo ω_0 of the scattering medium at each interaction. Sample weights may similarly become small after reflection from a surface with small, but non-zero, reflectance, or after multiple scattering in atmosphere with $\omega_0 < 1$; in either case small sample weights make small contributions to the overall radiative transfer calculation. Russian roulette is a technique in which samples with small w are re-sampled by adjusting w according the value of a uniform random number r, i.e.

$$w' = \begin{array}{l} 0 \text{ for } r \geq w/W \\ W \text{ for } r < w/W \end{array} \tag{1}$$

(see equation 6 in Iwabuchi 2006, and references therein). Normally $W = 1$. Time savings result because the sample terminates when $w' = 0$. Unlike simple truncating very weak samples (e.g. section 5a of Barker et al. 2003), Russian roulette does not introduce bias, but it decreases the time to solution by an amount depending on the optical depth of the medium, the single scattering albedo, and the surface albedo. Panel c of figure 2 shows the ratio of the cost of radiation transport (e.g. flux calculations) computed using Eq. 1 to the cost of of the same calculation without using Russian roulette. Savings at small values of optical thickness in this example arise because samples reflected from the dark surface (albedo 0.05) have small weights, so that most are are terminated by Eq. 1; if the underlying surface were absorbing or perfectly reflecting no time would be saved in optically thin media.

Iwabuchi (2006) adapted Russian roulette for use during using local estimation. At each scattering event the local estimate may be small if either a) the phase function at the angle between the direction of travel and the direction of the intensity is small, or b) if the integrated extinction between the scattering location and the medium boundary is large; in both cases small contributions are replaced with Russian roulette estimates (see Eq.s 13 and 14 in Iwabuchi 2006) . This reduces the number of paths that must be traced from scattering locations to the medium boundary during local estimation. The cost of intensity calculations using this algorithm, relative to the cost of radiation transport (e.g. flux calculations), is shown in the dashed lines in panel b of 2; considerable time is saved relative to equivalent calculations that do not use Russian roulette during local estimation, particularly at large optical depths and low viewing zenith angles. The method, like Eq. 1, is unbiased.

d. Methods for reducing variance in intensity calculations

At visible wavelengths the scattering phase function for cloud drops has a strong forward peak due to diffraction. This poses a particular problem for local estimation: collisions in which the sample is oriented in one of the directions at which intensity is desired are very rare but the phase function is so large that just a few collisions can dominate the intensity estimate. If the number of samples per batch is moderate these poorly-sampled events can lead to large values of standard error in intensity estimates. This problem is resolved when averaging over very large numbers of samples but the requisite amount of simulation can be computationally daunting if pixel-by-pixel intensities are desired.

1) Modifying the scattering phase function used for intensity calculations

One alternative is to modify the calculation, typically by modifing the scattering phase function used for local estimation (see, e.g. Antyufeev 1996; Barker et al. 2003; Iwabuchi 2006). This reduces the variance between batches but may introduce biases in intensity estimates. (The phase functions used during transport are not affected, so estimates of flux and flux divergence are unchanged.) As one such variance-reduction method, the I3RC Community Monte Carlo model provides the ability to replace the scattering phase function used for local estimation with a hybrid

$$\tilde{P}(\cos\Theta) = \begin{array}{l} P_G(\Theta) \text{ for } \Theta \leq \Theta_m \\ \\ P(\Theta) \text{ for } \Theta > \Theta_m \end{array} \tag{2}$$

where $P(\Theta)$ is the original phase function, Θ_m the matching angle, and $P_G(\Theta)$ a Gaussian. User specify the width of the Gaussian σ_G and the value of Θ_m is chosen so that $P(\Theta_m) = P_G(\Theta_m)$. The value of Θ_m is roughly commensurate with σ_G but its exact value depends on the magnitude of the forward peak in the original phase function. (In the calculations described in Section 4, for example, Θ_m/σ_G lies between 1.4 and 1.8 for cloud effective radii between 1.5 and 18 μm.) The Gaussian phase function is normalized to ensure that

$$\int_{\cos(\Theta_m)}^{1} P_G(\cos\Theta) d\cos\Theta = \int_{\cos(\Theta_m)}^{1} P(\cos\Theta) d\cos\Theta \tag{3}$$

The original phase function $P(\Theta)$ can be used for local estimation during the first N_e orders of scattering.

Any method which modifies the phase function for local estimation will have trade-offs between decreased standard error and increased absolute error. Figure 3 shows these relationships for Eq. 2 in homogeneous media using a scattering phase function appropriate for a distribution of cloud drops with effective radius 8.8 μm. In these calculations the sun is at 60 degrees; intensity errors are computed as the root-mean-squared error over 31 viewing directions in the principal plane from zenith to 75 degrees in the forward- and back-scattering directions. The decrease in standard error is expressed as the factor by which run time would need to be increased to make the standard error in the unmodified calculation match the standard error using 2, i.e. by the square of the ratio of the standard errors. Standard error decreases steeply and error increases slowly as σ_G increases and the full phase function is modified more dramatically, though the error increases for smaller optical thickness as low order scattering contributions become more important. Using Eq. 2 only when $N_e > 2$ (dashed lines) decreases the error by a factor of 2 or more but also decreases the amount of variance reduction, particularly at small optical thicknesses where low orders of scattering dominate the local estimate. Both error and the amount of variance reduction are influenced by the surface albedo at low optical depths, and would approach zero if the underlying surface were black.

2) REDISTRIBUTING INTENSITY IN SPACE

The I3RC Community Monte Carlo Model also implements a novel method introduced by Barker et al. (2003) for reducing the variance of intensity estimates by smoothing the estimate in space. Especially large contributions during the local estimation are trunctated. The excess is accumulated and, at the end of the calculation, distributed throughout the domain in an amount proportional to the intensity itself. This acts to reduce the pixel-to-pxiel variance in intensity. Domain-mean intensity is unaffected, though the method tends to reduce the highest values of intensity and enhance the lowest values (Barker et al. 2003), with the result that the error characteristics for spatially-resolved intensity depend strongly on the problem being solved.

3. SHDOM as a parallel explicit method

In principle the term SHDOM (Spherical Harmonics Discrete Ordinates Method; see Evans 1998) describes an algorithm; in practice, it also describes a freely-available code that implements the algorithm. SHDOM represents the radiation field by discretizing the angular dependence with a spherical harmonics series and the spatial dependence on a Cartesian grid. The radiative source function is stored as a variable-length series of spherical harmonics. The radiative transfer equation is solved iteratively by 1) transforming the source function from spherical harmonics to discrete ordinates, 2) integrating the product of the source function and transmission along discrete ordinates, 3) transforming the discrete ordinate intensities to spherical harmonics, and 4) calculating the source function from the intensity in spherical harmonics space. To achieve higher accuracy with a limited amount of memory, an adaptive grid is implemented in which regions where the source function is changing more rapidly have a higher density of grid points. The horizontally-uniform ”base grid” on which the problem is defined is supplemented as required by adaptive grid points formed by splitting cells in half during the solution iterations.

a. Reformulation of SHDOM as a parallel algorithm

SHDOM has been re-formulated as a parallel code. Parallelism is achieved using the Message Passing Interface (MPI; see, for example, `https://computing.llnl.gov/tutorials/mpi/`) which supports both shared- and distributed-memory environments. Parallel calculations require a horizontally periodic domain. Using SHDOM in a parallel mode can decrease the wall-time to solution but, perhaps more importantly, it allows for the solution to much larger problems than the serial version, since the memory required to represent the radiation field may be distributed among processors.

The computational domain is divided among processors with base grid points repeated on both sides of each boundary. Data are passed between neighboring processors during three steps in the algorithm: 1) the direct beam solar flux calculation at each internal grid point, 2) the radiative transfer equation integrations along each discrete ordinate during the solution iterations, and 3) the integrations along the viewing directions for the intensity output. Data transfer to and from files at the beginning and end of the calculation are handled by a single processor.

The majority of data transfer occur during the solution iterations (step 2 in the previous paragraph). During those iterations the discrete ordinate integrations proceed backwards from each grid point until a cell face is reached for which the intensities are known at all four corner grid points. When using a single processor the backwards integration usually stops after just one grid cell due to the discrete ordinate octant dependent ”sweeping” order. For each horizontal plane of grid points, however, there is one line of grid points for which the integration must travel back to the previous plane to reach grid points with known intensities. For multiple processors, the ”upwind” grid points on each processor sub-domain boundary are calculated (for one ordinate direction and one horizontal plane of base grid points at a time) with a special routine before the rest of the integration proceeds. This routine first carries the integrations from the entering boundary to the exiting boundary. If a cell face with known intensities is reached the completed intensity is stored to send back to the originating processor. Otherwise, the exiting position, accumulated transmission, and partial intensity integral are sent to the neighboring processors. This intensity integration and information passing is iterated until the discrete ordinate integrations have finished on all processors, when sub-domain boundary intensity values are exchanged with all the other processors. Direct solar beam transmission integrations and the output intensity integrations along the viewing directions are carried out in a similar manner.

The parallel implementation of SHDOM does not give exactly the same results as the serial version due to differences in the discrete ordinate integrations in the iterative solver which may be amplified by the discrete nature of the cell splitting decisions for the adaptive grid. Differences between parallel and serial solutions are small, however, compared to the overall accuracy of the algorithm.

b. Efficiency of SHDOM on multiple processors

The parallel version of SHDOM can shorten the wall-time to solution but it increases the total amount of calculation required. There are three sources of extra computation: the repeated grid points at the edge of each sub-domain; the time processors spend passing data to each other or waiting for other processors to synchronize; and the extra effort (described in the last section) required for tracing the discrete ordinate intensities. All sources of overhead are reduced (in a fractional sense) by making the sub-domains as large as possible, so that the ratio of edge points to interior points is small.

The amount of time used by SHDOM running on multiple processors relative to a single processor depends on the problem being solved, the number of processors used, and various aspects of the computational architecture. Figure 4 shows results for two widely-available architectures: a shared-memory system using two processors with four cores each and a distributed-memory supercomputer. Radiative transfer is computed for the stratocumulus cloud field described in the next section using a base grid of $128 \times 128 \times 41$ points and resolution parameters taken from run 4F in Table 2. Figure 4 shows the ratio of the total CPU time used, relative to that used by a single processor, as a function of the number of processors used. (Uncontrollable issues cause small random variations in this ratio.) For both architectures the total CPU time used increases significantly above some number of processors as the additional costs described in the last paragraph become noticeable. Note that the algorithm used for domain decomposition minimizes the number of boundary points when the number of processors is an integer squared (when the domain is mapped as $\sqrt{N_p} \times \sqrt{N_p}$ processors), and maximizes the number of boundary points when the number of processors is a prime (when the domain is mapped as $N_p \times 1$ processors).

Processor loads may become unbalanced if the optical properties in the sub-domains, and hence the density of internal grid points, varies significantly. To use computational resources more efficiently a load balancing procedure may be enabled. This procedure uses the number of base and adaptive grid points from a fast, low angular resolution run to adjust the sub-domain boundaries so that the number of points is roughly the same among processors. Load balancing can also be used to account for non-uniform processor speeds.

4. Two prototype problems

We illustrate the solution characteristics of explicit and Monte Carlo methods using two cloud fields produced by large-eddy simulations as prototype problems. One is a 3.5 km square domain, 1 km deep, filled with stratocumulus clouds; this scene was used in Phase II of the I3RC inter-comparisons (Cahalan et al. 2005). The second is a field of scattered cumulus over the ocean from simulations made by Stevens and Seifert (2008). The original domain is 3 km deep and 19.2 km on a side; in these calculations we use a 4.8 km square subset of this domain chosen to minimize the number of cloudy pixels along the edge. This sub-domain has a vertically-projected cloud fraction of 14.8%. Figure 9 shows images of these fields from high-resolution calculations with SHDOM and the I3RC Monte Carlo model.

For each cloud field we perform simulations at a non-absorbing wavelength in the visible (0.67 μm), and at a mildly absorbing wavelength in the near infrared (2.13 μm). To convert cloud water content to optical properties we assume that cloud water is divided into droplets whose sizes obey a gamma distribution with effective variance 0.1. Effective radius in each is diagnosed from the liquid water content assuming constant droplet number (72/cm^3 for the stratocumulus, per the I3RC case specification, and 70/cm^3 for the cumulus, per the simulation) with a minimum size of 4 μm. We pre-compute single scattering properties (extinction cross-section, single scattering albedo, and scattering phase function) as a function of drop size using Mie theory and tabulate these values at 50 equally spaced values of effective radius from 0.5 μm to 25 μm. Very few cells have values of effective radius that exactly match one of the tabulated values, so we interpolate the extinction and single scattering albedo and choose the phase function for the closest tabulated value of effective radius.

The cumulus domain includes aerosols. We use properties obtained from the Aeronet station on Cape

San Juan during Dec 2005 and Jan 2006. We use the highest optical aerosol optical depth observed during this period, namely $\tau_{\text{aerosol}} = 0.233$ at 0.675 microns; at this time the Aeronet retrievals, when mapped onto a log-normal size distribution, imply a an effective raduis $r_e = 0.964$ microns, log-normal standard deviation of 1.10, and complex index of refraction $m = 1.4 - 0.0029i$. We assumed a complex index of refraction of $m = 1.36 - 0.0020i$ at 2.13 microns; the optical thickness at this wavelength is $\tau_{\text{aerosol}} = 0.2$. The aerosols are distributed uniformly throughout the 3 km deep domain. Rayleigh scattering is also included in the cumulus cloud fields for the 0.67 μm simulations.

For all calculations the sun introduces radiation in the direction $\mu_0 = -0.707, \phi = 35$. The flux is normalized to 1 on a horizontal plane. The surface is Lambertian with albedo 0.05. We compute upwelling flux at the top of the domain, downwelling flux at the bottom of the domain, and, for the calculation at 2.13 microns, flux divergence (proportional to heating rate) within the domain. Intensity is computed at nine angles symmetric about nadir in the $\phi = 0$ plane, corresponding to the viewing angles of the MISR instrument, i.e. at nadir and at μ = (0.334, 0.5, 0.7, 0.898) for $\phi = 0$ and $\phi = 180$.

5. Accuracy and standard error using Monte Carlo variance reduction methods

Section 2.d described variance-reduction methods for Monte Carlo methods that may be used to decrease the standard error of the calculation at the cost of increased bias. Both the amount of variance reduction and the bias introduced depend in part on the aggressiveness with which the techniques are applied, such that larger values of σ_G and smaller values of N_e in Eq. 2, or smaller values of the maximum allowed local estimate ζ_{max} (for the method described in Sec. 2.d.2), tend to increase bias and make the results less variable. But both cost (increased bias) and benefit (decreased variance) also depend on the relative frequency with which large contributions to the local estimate are encountered, since this determines how effective the reduction can be; this depends in turn on the distribution of optical properties within the domain and the geometry of the problem being solved (i.e. the angle between incoming solar radiation and each of the angles at which intensity is computed). Thus neither cost nor benefit can be predicted *a priori* and the choice of parameters is necessarily somewhat subjective.

We explore the performance of the two variance reduction methods described in Sec. 2.d in both prototype scenes over a range of parameters detailed in Table 1. The runs are specified more fully in Section 7; most relevant here is that calculations using $10^n, n = (6, 7, 8, 9)$ samples are compared to benchmark results using 10^{10} samples.

Both variance reduction methods reduce the error in intensity calculations to some degree at most resolutions (see Figure 5). On its own the spatial redistribution of large local estimates (run a) is more effective in the stratocumulus scene than in the cumulus where the cloud fraction, and hence the area over which excess contributions can be spread, is small. Spatial redistribution alone has less effect when the number of samples (and the computation time) is large. Using any form of the hybrid phase function (Eq. 2) decreases the error by a factor of three or more, which is roughly equivalent to computing ten times as many samples. More aggressive settings are more effective at low sample numbers but the error decrease becomes less sensitive to the parameters when 10^9 trajectories are used.

But this reduced error comes at a cost. Monte Carlo estimates of uncertainty, as measured by the standard error of multiple sub-calculations, are almost perfect estimates of the true RMS error when unbiased methods are used (i.e. the circle in Figure 6 lie on the 1:1 line) but this agreement is lost when variance reduction methods are used (letters in Fig. 6). This occurs when the Monte Carlo noise becomes smaller than the bias introduced by modifying the phase function or local estimate; in these calculations this is evident in all of the highest trajectory-count (lowest error) calculations and, for some parameter sets, for runs using 10^8 samples. (The standard error of the benchmark runs is 3-30 times smaller than the RMS of the calculations using variance reduction, so the disagreement can not be attributed to sampling error.) Whether these small biases are acceptable or not depends on the application.

6. Resolution, cost, and accuracy using SHDOM

For any given problem the computational cost and accuracy of the SHDOM algorithm depend primarily on the angular and spatial resolution used in the simulation. Calculations are most efficient when these resolutions are well-matched. We illustrate this point using a series of calculations at 0.67 microns for the two prototype problems described in the previous section.

SHDOM angular resolution is specified by the number of discrete ordinates in zenith angle (N_μ) and azimuthal angle (N_ϕ); often these are chosen to ensure isotropic angular resolution, i.e. $N_\phi = 2N_\mu$. Spatial accuracy is specified by the number of grid points in the base grid ($N_x \times N_y \times N_z$, $N_y = N_x$) and by the splitting accuracy, which controls how many adaptive grid points are produced. We perform simulations using six values of angular resolution parameters (numbers 1 to 6) and eight values of spatial resolution, five (sets A-E) in which the base grid matches the grid on which optical properties are defined and three (sets F, G, and H) in which the base grid is roughly twice as dense in the horizontal. This results in 48 simulations for each problem. Simulations are divided among a number of processors approximately proportional to the memory required; this numbers ranges from 1 to 64. The solution accuracy is 10^{-5} and sequence acceleration is used during solution iterations. Load balancing is not used: the stratocumulus field is relatively uniform, while the low cloud fraction of the cumulus field results in a small proportion of adaptive grid points. (Using load balancing for the cumulus field at resolution 5E, for example, resulted in a 10% CPU time improvement for 16 processors.)

Accuracy is assessed by comparison to the run with the highest spatial and angular resolution (run 6H). Figure 7 shows the accuracy in intensity calculations for both scenes, measured as the root-mean-square error over every grid point and every angle; Figure 8 shows analogous results for upwelling and downwelling fluxes at the top and bottom of the stratocumulus scene.

For the stratocumulus scene, errors in intensity decrease rapidly with decreasing splitting accuracy for any given angular resolution, then remains roughly constant as angular resolution dominates the error. Error increases at resolution F relative to E at higher angular resolutions because the splitting accuracy relaxes (from 0.02 to 0.03) while the base grid resolution increases.

For the cumulus scene, adaptive cell splitting substantially improves accuracy in intensity calculations at little computational cost for high angular resolution ($N_\mu \geq 16$) calculations. This is because cells are split only in the small fraction of the domain that is cloudy. Doubling the spatial resolution of the base grid (runs F-G) results in a five-fold increase in computational cost with almost no gain in accuracy; this is also true for flux (not shown).

The relationships between resolution, cost, and error in calculations of upwelling flux for the stratocumulus scenes (Fig. 8) are qualitatively similar to those for intensity, although the base grid resolution is more important. This is a result of how the algorithm operates: SHDOM calculates grid point fluxes from the discrete ordinate intensities during the solution iteration, so the base grid resolution in clear sky is important to the flux accuracy at the top and bottom of the domain because there are no adaptive grid cells in the clear areas. Intensity is calculated after the solution iterations by integrating the source function, so the clear sky is not important, and accuracy in intensity is controlled by the splitting accuracy.

Accuracy in downwelling flux at the surface, though, is almost entirely controlled by angular resolution, and increases in spatial resolution only increase the computational time. This is because the downwelling radiation has more angular structure (from low orders of scattering of the direct solar beam), and so requires higher angular resolution to capture, than does upwelling radiation. Diffuse radiation at cloud base is much less spatially variable than at cloud top, however, so spatial resolution has little influence on accuracy.

7. Relative efficiency of Monte Carlo and SHDOM

In this section we address a practical question: given a choice between SHDOM and the I3RC model, which code provides the most efficient path to solution of a given radiative transfer problem? The more efficient model, in this context, is the one that produces the lowest error for a given computational cost. We answer this question by performing identical calculations with both models at a range of resolutions and comparing the relationships between error and computational cost for a range of quantities.

We select a subset of the SHDOM resolution parameters described Table 2, namely resolutions (1A, 2B, 3C, 4D, 5G) for the stratocumulus case and (1C, 2C, 3B, 4C, 5D) for the cumulus case. For both sets of calculations resolution 6H provides the benchmark. Monte Carlo calculations use ray tracing for radiation transport and employ Russian roulette re-weighting of weak samples. Results for flux and flux divergence use relatively fast simulations in which intensity is not computed. Russian roulette is also employed in intensity calculations. Monte Carlo resolution is essentially determined by the number of trajectories. For flux calculations we compare results for $10^n, n = (6, 7, 8)$ samples with a benchmark calculation using 10^9 samples; benchmarks for intensity calculations use 10^{10} samples.

Example benchmark intensity calculations are shown in Fig. 9. The benchmark calculations agree to within a few tenths of a percent for most quantities (see Table 3). Some of the remaining differences are rooted in the algorithms themselves. Though we have specified the same distribution of optical properties for both radiative transfer models, SHDOM and the I3RC Monte Carlo model solve slightly different problems: SHDOM assumes that the medium varies linearly between cell corners, while the Monte Carlo model assumes that cells are internally uniform with discontinuous jumps at cell boundaries.

Figure 10 shows the error in flux calculations, measured as the root-mean-square over all pixels normalized by the domain mean, as a function of total time to solution for both models and both scenes. Both axes are logarithmic so the Monte Carlo error, which decreases as the inverse square root of the solution time, appears as a straight line. Monte Carlo standard error estimates (not shown) agree almost perfectly with the true error. Both models produce larger errors for the cumulus scene than for the stratocumulus because the domain is larger. SHDOM becomes more efficient than Monte Carlo in computing upwelling fluxes for the cumulus scene as the resolution increases, but is increasingly less efficient for the stratocumulus case. Figure 11 shows the relationships between computational time and the error in flux divergence calculations at 2.13 μm, measured as the RMS over all cells in the domain and normalized by the domain-mean profile. Absolute values of flux divergence are much smaller in the cumulus scene than in the stratocumulus, so both models produce lower errors in the cumulus field.

The two models produce roughly comparable errors in spatially-resolved flux and flux divergence for a given computational cost. The cell-splitting employed by SHDOM is a particular advantage in the cumulus field because grid resolution is high in a small portion of the domain. SHDOM errors in downward flux decrease slowly with computation time in the stratocumulus cloud field because there is no way except high angular resolution to resolve the angular dependence of the downwelling flux.

For a given time to solution, accuracy for pixel-level fluxes for SHDOM and Monte Carlo methods are within a factor of ten, but for pixel-level intensity calculations SHDOM is much more efficient. Figure 12 shows the error in these calculations, expressed as the root-mean-square over all pixels in all nine directions and normalized by the domain-mean independently in each direction, for both models. Monte Carlo results are provided for unbiased calculations and for an example computed using Eq. 2 with parameters from run D in Table 1 ($\sigma_G = 5^\circ$ and $N_e = 3$). Using the hybrid phase function reduces the error substantially but, as we showed in Fig. 6, estimates of the uncertainty (downward-pointing triangles) are biased at the highest resolutions relative to the true error (upward-pointing triangles). Our choice of parameters from Table 1 was conservative but even the most aggressive application of variance reduction can not reduce the error for a given computational cost to those approaching SHDOM.

Errors in Monte Carlo estimates of domain-average intensity are smaller by a factor of $1/\sqrt{N_{x,y}}$, where $N_{x,y}$ is the number of pixels (64^2 for the stratocumulus scene and 96^2 for the cumulus), while errors for

SHDOM the errors remain roughly constant. This implies that the two methods are about equally efficient at computing domain-averaged intensity.

Time to solution is not the only computational cost; the amount of memory required may also be a constraint. Explicit methods such as SHDOM must store the complete five dimensional radiation field which can be a significant burden at high resolution. Figure 13 compares the memory and computation time used by SHDOM and Monte Carlo for each of our two scenes. SHDOM memory use increases steadily with CPU time, covering a range of almost 500 for the stratocumulus scene resolutions used in Fig. 13. Note that if aerosols are omitted from the cumulus scene the adaptive spherical harmonic truncation in SHDOM becomes effective and memory use drops by a factor of 3.5. For the highest resolution calculation on the cumulus scene SHDOM uses 71Gb of memory, demonstrating that SHDOM resolution may be limited by memory constraints, especially on smaller computers, long before computation time becomes an issue.

Monte Carlo memory use in Fig. 13 is for runs in which intensity is calculated; runs in which only radiation transport is computed use about 15-30% of this amount. For each scene, increases in memory use reflect only the number of processors used in the calculation, which ranged from 4 to 44; this number was chosen to make the most efficient use of supercomputer resources (i.e. we used the smallest integer multiple of 4 processors which would allow the calculation to finished in less than 8 hours). In principle, however, all Monte Carlo calculations could be done on a single processor using the amount of memory indicated by the horizontal lines, trading decreased memory use for an increase in the elapsed time to solution.

8. Matching tools to tasks

Either explicit or Monte Carlo methods can be used to solve a given three-dimensional radiative transfer problem and the methods are in very good agreement (see, e.g., Table 3) when sufficient computational resources are available. But even low-resolution, high-error calculations are very computationally expensive, so identifying the most efficient solution method may be important. The results in Section 7 provide some guidance in choosing the most efficient tool for a given problem.

SHDOM and Monte Carlo methods are roughly equally efficient at computing pixel-level fluxes and cell-by-cell flux divergence. SHDOM benefits when computing upward flux in a medium where optically thick regions ($\tau > 1$) are sparse, as in our cumulus example. Monte Carlo is somewhat superior for computing downwelling flux, since SHDOM requires high angular resolution to resolve the low orders of scattering. Because Monte Carlo errors are uncorrelated in space (i.e. between pixels) the Monte Carlo method is much more efficient at computing domain-average fluxes and flux divergence profiles than SHDOM.

SHDOM is vastly more efficient than Monte Carlo methods for computing pixel-by-pixel intensity when intensity in multiple directions is required. Accuracy improvements from variance reduction techniques are significant but modest compared to this difference. SHDOM and Monte Carlo are comparably efficient at computing domain-average intensity in a range of directions. In the special case that a single intensity is computed, particularly in the nadir direction, the difference is less stark (see Fig. 2).

These results apply to quasi-monochromatic calculations. Many applications, however, require integration over some portion of the electromagnetic spectrum. Two common examples are the calculation of broadband fluxes for energy balance problems and the calculation of narrowband intensities for remote sensing problems. The latter calculations are typically made in narrow spectral regions (”bands”) in which the optical properties of most constituents can be considered constant, requiring only the use of a k-distribution to treat gaseous absorption within the band. Broadband calculations require calculations in a series of bands.

Monte Carlo methods can include spectral integration within or across bands at very little cost by dividing samples among spectral intervals, while SHDOM must perform a series of independent calculations. (To avoid uncommon but severe errors SHDOM has been changed and now uses an independent initialization for each k in a distribution.) SHDOM’s convergence rate increases with the amount of absorption, so that computation time increases less than linearly with the number of ks. Assuming even $O(10)$ ks per band SHDOM remains much more efficient for remote sensing problems requiring pixel-scale intensity values. Monte Carlo methods can become comparable if substantial spatial averaging of the pixel-level narrow-band intensities was required (e.g for computing the signal at a remote sensor with spatial resolution of order kilometers), and they are are substantially more efficient for broadband problems. Since the two models are comparably efficient at computing monochromatic grid cell flux divergence, Monte Carlo methods are therefore much more efficient at computing the broadband flux divergence needed for coupling to dynamical models.

These generalizations apply when the grid spacing is smaller than the radiation mean free path., i.e. when the 3D radiative transfer is resolved. On the coarser resolution grids such as those typical of cloud-resolving models, SHDOM must employ significant cell-splitting, so that Monte Carlo methods may become substantially more efficient at computing pixel-by-pixel fluxes.

Efficiency may not be the only consideration in choosing a model, especially where the two methods are comparable. Monte Carlo methods are particularly simple to use: a single parameter controls accuracy and the uncertainty in (unbiased) calculations is known. Alternatively, calculation can be performed until some specified accuracy is reached. In this way Monte Carlo techniques requires less sophistication on the part of the user.

Acknowledgments

We appreciate the support, suggestions, and enthusiasm of the I3RC community. Alexander Marshak helped us clarify some points regarding Monte Carlo methods, and we thank Dr. Olga L. Mayol-Bracero for her effort in establishing and maintaining the Cape San Juan Aeronet site. RP was supported by NASA under awards NAG5-11279 and NNG04GM91G and by the National Science Foundation Science and Technology Center for Multi-scale Modeling of Atmospheric Processes, managed by Colorado State University under cooperative agreement No. ATM-0425247. KFE was supported by the NASA Radiation Sciences Program under award NNX07AQ84G.

REFERENCES

Antyufeev, V., 1996: Solution of the generalized transport equation with a peak-shaped indicatrix by the monte carlo method. *Russ. J. Numer. Anal. Math. Model.*, **11 (2)**, 113–137.

Barker, H. W., R. K. Goldstein, and D. E. Stevens, 2003: Monte Carlo simulation of solar reflectances for cloudy atmospheres. *J. Atmos. Sci.*, **60 (16)**, 1881–1894, doi:10.1175/1520-0469(2003)060⟨1881:MCSOSR⟩2.0.CO;2.

Cahalan, R. F., et al., 2005: The I3RC - Bringing together the most advanced radiative transfer tools for cloudy atmospheres. *Bull. Amer. Meteorol. Soc.*, **86**, 1275+, doi:10.1175/BAMS-86-9-1275.

Di Giuseppe, F. and A. M. Tompkins, 2003: Three-dimensional radiative transfer in tropical deep convective clouds. *J. Geophys. Res. - Atmos.*, **108 (D23)**, 4741.

Evans, K. F., 1998: The spherical harmonics discrete ordinate method for three-dimensional atmospheric radiative transfer. *J. Atmos. Sci.*, **55 (3)**, 429–446, doi:10.1175/1520-0469(1998)055⟨0429:TSHDOM⟩2.0.CO;2.

Evans, K. F. and A. Marshak, 2005: Numerical methods. *3D Radiative Transfer in Cloudy Atmospheres*, A. Marshak and A. B. Davis, Eds., Springer, 243–282.

Iwabuchi, H., 2006: Efficient Monte Carlo methods for radiative transfer modeling. *J. Atmos. Sci.*, **63 (9)**, 2324–2339, doi:10.1175/JAS3755.1.

Loeb, N. G., T. Varnai, and D. M. Winker, 1998: Influence of subpixel-scale cloud-top structure on reflectances from overcast stratiform cloud layers. *J. Atmos. Sci.*, **55 (18)**, 2960–2973.

Marchuk, G., G. Mikhailov, M. Zazaraliev, R. Darbinjan, B. Kargin, and B. Eleopv, 1980: *The Monte Carlo methods in atmospheric optics*. Springer-Verlag, New York, NY.

Marshak, A., S. Platnick, T. Varnai, G. Y. Wen, and R. F. Cahalan, 2006: Impact of three-dimensional radiative effects on satellite retrievals of cloud droplet sizes. *J. Geophys. Res. - Atmos.*, **111 (D9)**, D09 207, doi:10.1029/2005JD006686.

Mishchenko, M. I., 2008: Multiple scattering, radiative transfer, and weak localization in discrete random media: Unified microphysical approach. *Rev. Geophys.*, **46**, RG2003, doi:10.1029/2007RG000230.

Scheirer, R. and A. Macke, 2003: Cloud inhomogeneity and broadband solar fluxes. *J. Geophys. Res. - Atmos.*, **108 (D19)**, 4599, doi:10.1029/2002JD003321.

Stevens, B. and A. Seifert, 2008: Understanding macrophysical outcomes of microphysical choices in simulations of shallow cumulus convection. *J. Meteorol. Soc. Japan*, **86A**, 143–162, doi:10.2151/jmsj.86A.143.

List of Tables

TABLE 1. Runs used to assess the performance of variance reduction methods for intensity in two prototype problems. Variables σ_G and N_e are defined in Eq. 2, while ζ_{max} is the maximum allowed value of the local estimate; we follow Barker et al. (2003) and set $\zeta_{\text{max}} = f(1+g)/4(1-g)^2$ with $g = 0.85$ and $f \approx 3$. Error is measured as the root-mean-square over all pixels and nine viewing angles relative to a benchmark calculation using 10^{10} trajectories. All variance reduction methods are biased estimators, meaning that standard error underestimates the true Monte Carlo sampling error.

Case	σ_G	N_e	ζ_{max}	Error ($\times 10^2$) using 10^9 trajectories			
	(deg)			Stratocumulus		Cumulus	
				RMS	Std Err.	RMS	Std Err.
O	–	–	–	3.01	2.91	0.48	0.46
a	–	–	75	3.02	1.19	0.50	0.33
b	5	–	–	1.22	0.73	0.25	0.20
c	9	–	–	1.21	0.50	0.23	0.14
d	5	3	–	1.39	1.04	0.29	0.25
e	9	3	–	1.31	0.90	0.27	0.21
f	5	3	75	1.23	0.75	0.27	0.22
g	9	3	75	1.14	0.53	0.25	0.18

TABLE 2. Angular and spatial resolutions used in SHDOM calculations.

			Stratocumulus		Cumulus		
Angular	N_μ	Spatial	N_x	N_z	N_x	N_z	Splitting accuracy
1	4	A	64	21	96	100	0.30
2	6	B	64	21	96	100	0.10
3	8	C	64	21	96	100	0.05
4	12	D	64	21	96	100	0.03
5	16	E	64	21	96	100	0.02
6	32	F	128	41	192	151	0.03
—	—	G	128	41	192	151	0.02
—	—	H	128	41	192	151	0.015

TABLE 3. Absolute differences between benchmark runs using SHDOM and the I3RC Community Monte Carlo model. Reflectance bias are the root-mean-square error over the nine directions at which intensity is computed. Most differences are less than 1%. Monte Carlo estimates of the standard error are show for comparison.

	Stratocumulus ($\times 10^3$)			Cumulus ($\times 10^3$)		
	Bias	RMS error	MC std. err.	Bias	RMS error	MC std. err.
Flux up	-3.72	5.00	1.26	1.21	1.64	0.62
Flux down	2.38	24.97	1.62	1.29	8.80	2.04
Reflectance	1.30	3.99	2.94	0.46	1.08	0.47

List of Figures

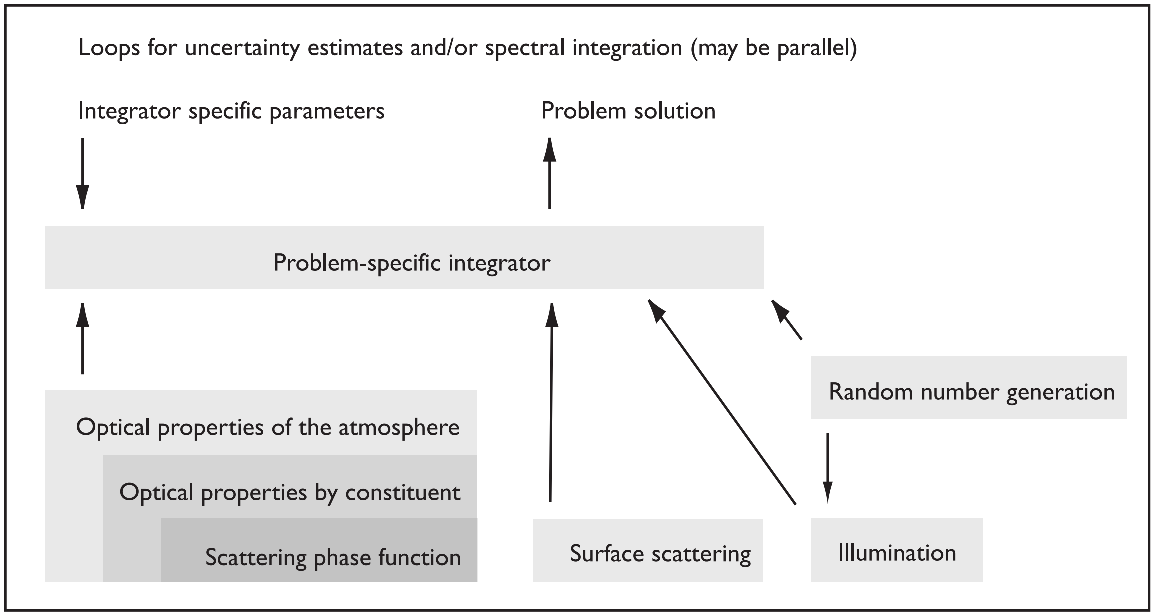
Loops for uncertainty estimates and/or spectral integration (may be parallel)
Integrator specific parameters
Problem solution
Problem-specific integrator
Random number generation
Optical properties of the atmosphere
Optical properties by constituent
Scattering phase function
Surface scattering
Illumination

FIG. 1. Software modules used to describe radiative transfer in a three-dimensionally varying atmosphere and the relationships of those module as used in the I3RC Community Monte Carlo model. Each box represents a distinct module; one box containing another indicates that the module represented by the outer box relies on the facilities provided by the inner box. Arrows indicate objects supplied as part of the problem definition or solution in the Monte Carlo model.

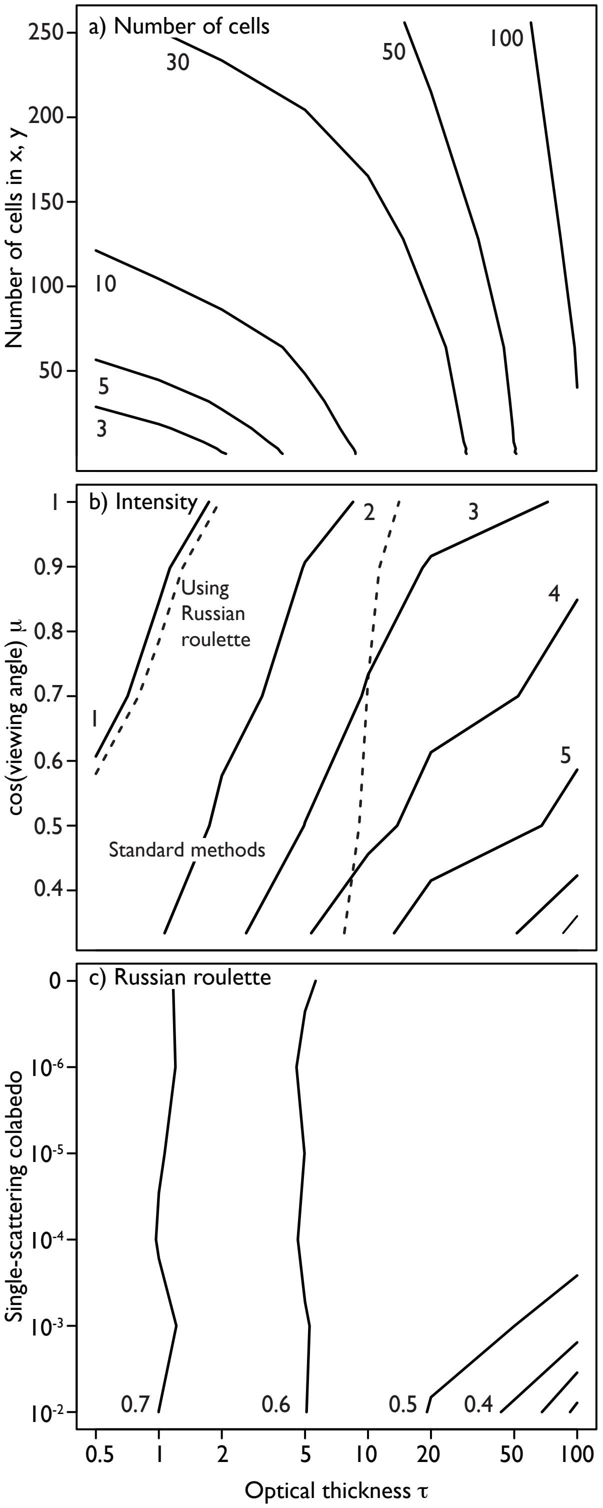

FIG. 2. Costs to compute radiative transfer using Monte Carlo methods in a plane-parallel medium as a function of optical thickness. a) Cost as a function of domain size relative to the cost for a single cell. Horizontal resolution is uniform in each dimension; there are one-fourth as many layers in the vertical as cells in each horizontal dimension. b) Additional time required to compute intensity as a function of viewing zenith angle, relative to the cost of the radiation transport calculation at that optical thickness. Solid lines represent the nominal cost; dashed lines show the cost using Russian roulette adapted for intensity by Iwabuchi (2006). c) Time to solution using Russian roulette during radiation transport (flux) calculations, relative to calculations without the method, as a function of single scattering co-albedo $1 - \omega_0$. Values less than one indicate time savings. Savings at small optical depths occur because samples reflect by the dark surface have small weights. In all panels the solar zenith angle is $\mu_0 = 0.707$, the surface is a Lambertian reflector with an albedo of 0.05, and scattering phase functions are representative of cloud droplets in the

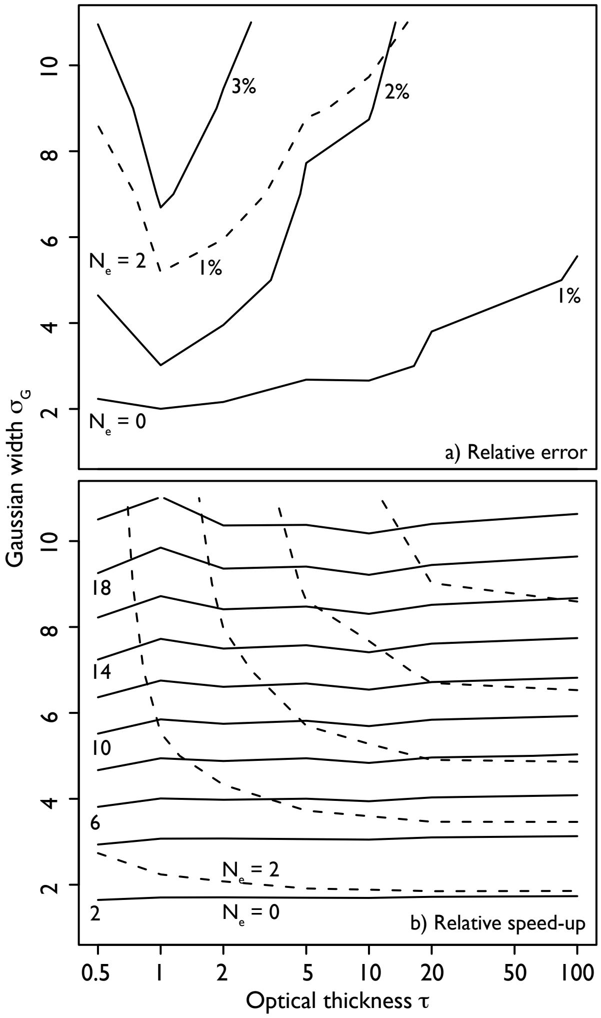

FIG. 3. Trade-offs between bias and amount of variance reduction using a hybrid phase function for local estimation in homogenous clouds. Solid lines indicate results for the phase function defined by Eq. 2; dashed lines show results when local estimates at the first two orders of scattering use the original phase function ($N_e = 2$). a) RMS normalized error in intensity over 31 equally-spaced upwelling viewing angles in the principal plane, as a function of the cloud optical thickness and the width σ_G of the Gaussian phase function used to blunt the highly-peaked original phase functions. Contours are every 1%, and the error for $N_e = 2$ lies in the range 0.5-1.5% everywhere. b) The amount of variance reduction achieved, expressed as the factor by which computation time for the unmodified time would need to be increased to reach this level of standard error. Contours are at factors of 2, 4, 6, ... Variance reduction can only be achieved at the cost of introducing bias, but a useful compromise may be possible for some applications.

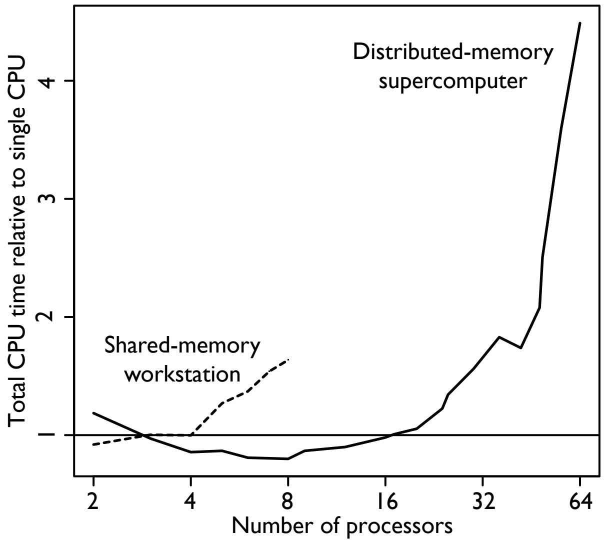

FIG. 4. Total computation time used by SHDOM as a function of the number of processors. The dashed line is taken from a two-processor, eight-core, shared-memory workstation; the solid line from a distributed-memory supercomputer. Cost is summed across all processors and is expressed relative to the amount used for a single processor. Scaling behavior is problem- and architecture-dependent, but degrades for large numbers of processors.

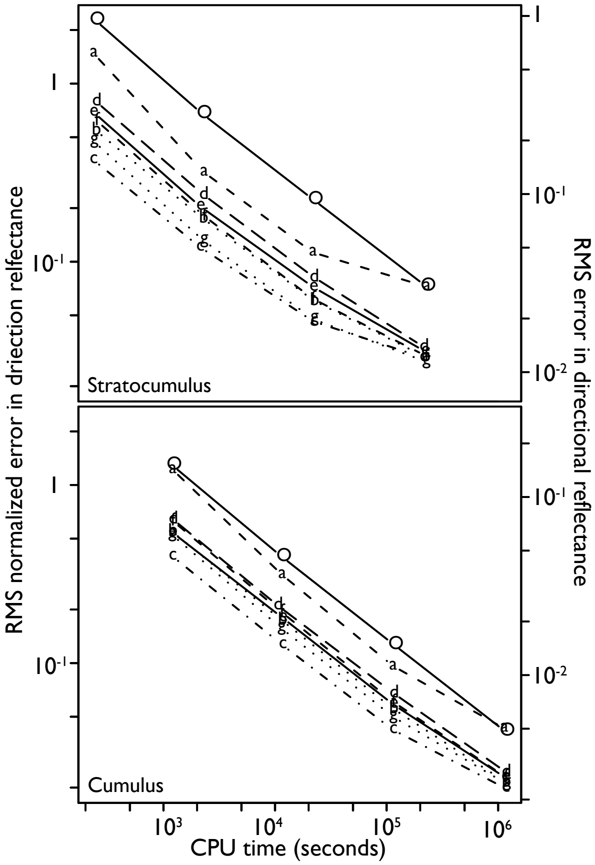

FIG. 5. Error in directional reflectance (intensity) calculations as a function of computation time for Monte Carlo methods with (letters a-g) and without (circles) variance reduction techniques. Errors on the left hand axis are normalized by the domain-mean value in each direction before taking the RMS over all directions and all pixels. Both axes are logarithmic. Letters refer to parameters sets detailed in Table 1. Slightly modifying the phase function used for local estimation (letters b-g) decreases error by roughly the same amount as increasing the number of trajectories by a factor of ten.

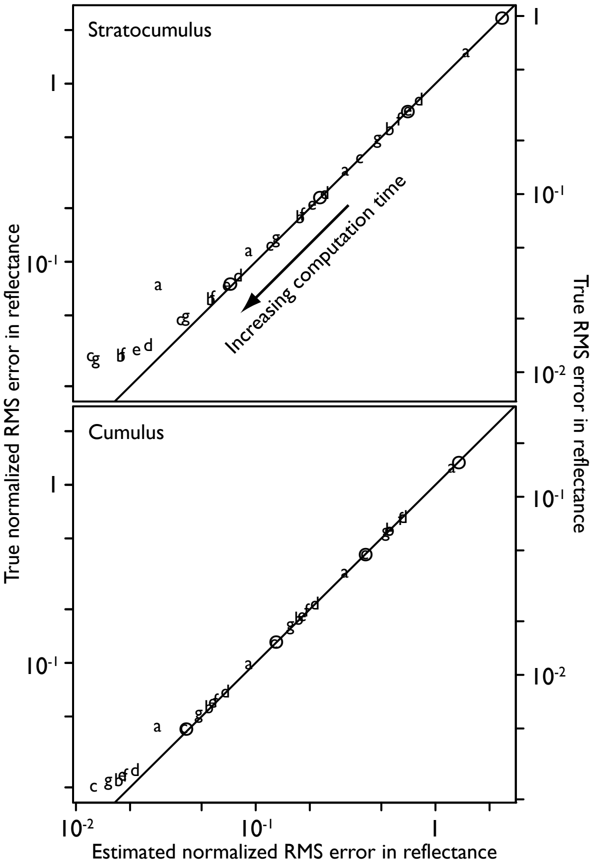

FIG. 6. True error in directional reflectance (intensity) calculations, estimated by comparison with an unbiased benchmark, as a function of standard error estimate for Monte Carlo calculations with (letters) and without (circles) the use of biased variance reduction methods. Both axes are logarithmic. The two estimates agree when the computation time is low and the Monte Carlo noise dominates the error, but when large numbers of trajectories are used the bias caused by modifying the local estimate calculation emerges.

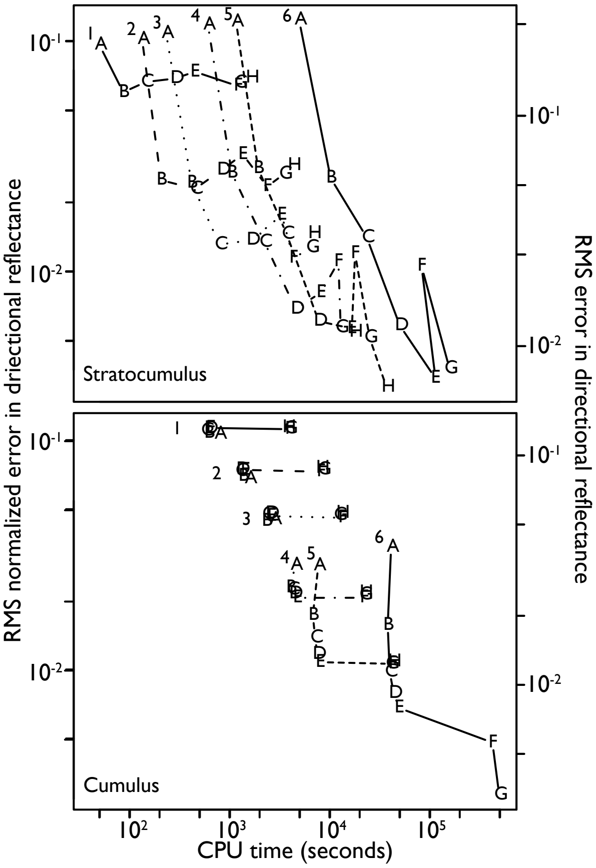

FIG. 7. Error in directional reflectance (intensity) computed using SHDOM as a function of total computation time for the stratocumulus and cumulus scenes. Both axes are logarithmic. Letters refer to spatial resolution and numbers (and line types) to angular resolution detailed in Table 2. Error is computed as the root-mean-square difference over all directions and all pixels from the benchmark (6H) calculation. Values on the left axis are computed by normalizing the error by the domain-mean in each direction before computing the RMS over directions; the right axis shows the RMS over direction of the raw error. When calculations in the cumulus scene use low angular resolution (cases 1-4), increased spatial resolution increases computation time without increasing accuracy, so points A-E are clustered to the left and F-H on the right.

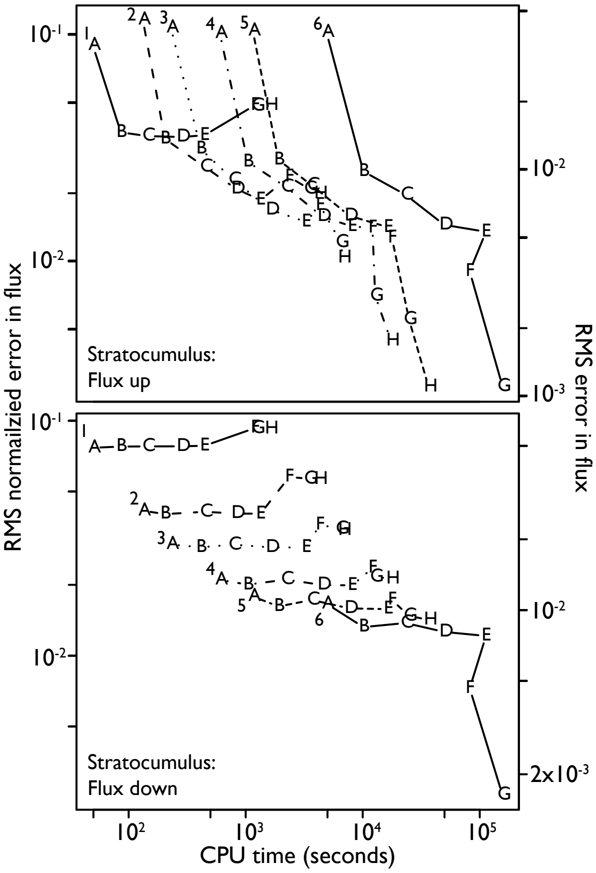

FIG. 8. Error in flux calculations, measured as the root-mean-square difference from the benchmark (resolution 6H) over all pixels, as a function of total CPU time in the stratocumulus scene for upwelling flux at the domain top and downwelling flux at the surface. The RMS flux difference normalized by the domain mean flux is shown on the left axis; absolute values on the right. Accuracy in downwelling flux depends mostly on angular resolution; accuracy in upwelling flux depends on spatial resolution as well.

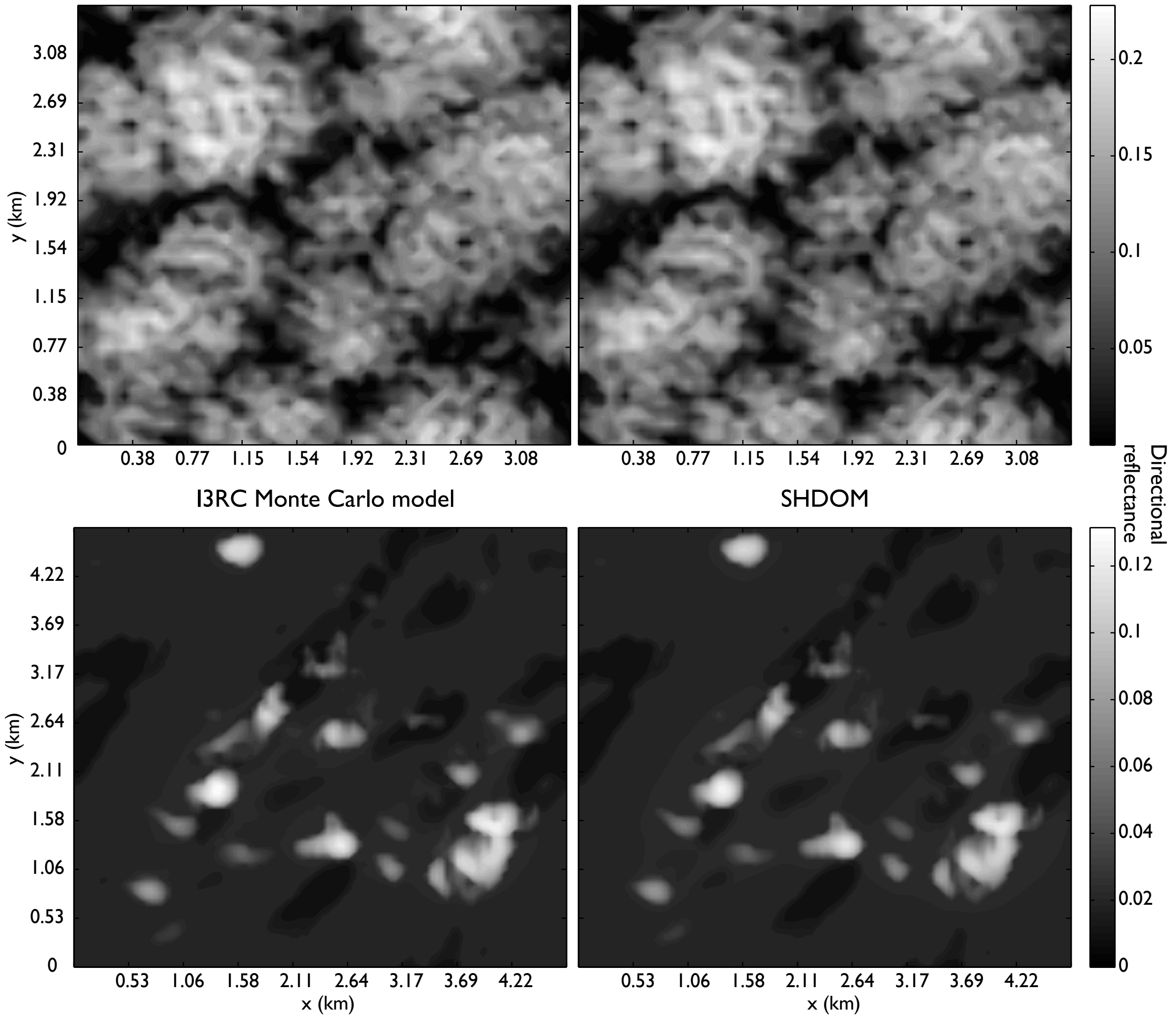

FIG. 9. Nadir reflectance (intensity times π) for the two scenes described in Sec. 4 computed with the I3RC community Monte Carlo model (left column) and SHDOM (right column). Differences between the two models are quantified in Table 3; in most cases they are a fraction of a percent of the incoming solar flux.

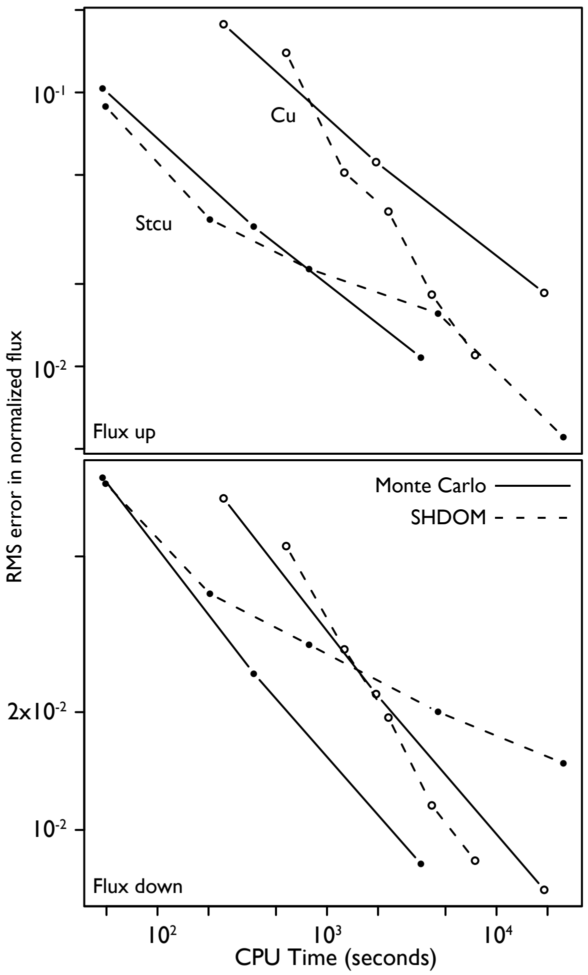

FIG. 10. Error in flux at 0.67 μm as a function of total computation time for calculations with the I3RC Community Monte Carlo model (solid lines) and SHDOM (dashed lines). The upper panel shows results for flux up at the top of the domain, the lower panel flux down at the domain base; in both panels filled symbols refer to the stratocumulus scene and open symbols the cumulus scene. Error is the root-mean-square over all columns in the domain, normalized by the domain mean, and assessed by comparison with a benchmark calculation. Monte Carlo results for flux use radiation transport calculations only.

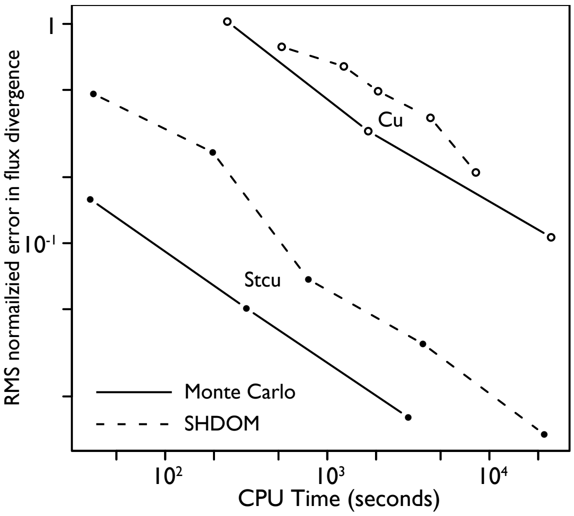

FIG. 11. Error in flux divergence at 2.13 μm as a function of total computation time for calculations with the I3RC Community Monte Carlo model (circles) and SHDOM (squares). Error is the root-mean-square over all cells in the domain, normalized by the domain mean, and assessed by comparison with a benchmark calculation. Both axes are logarithmic.

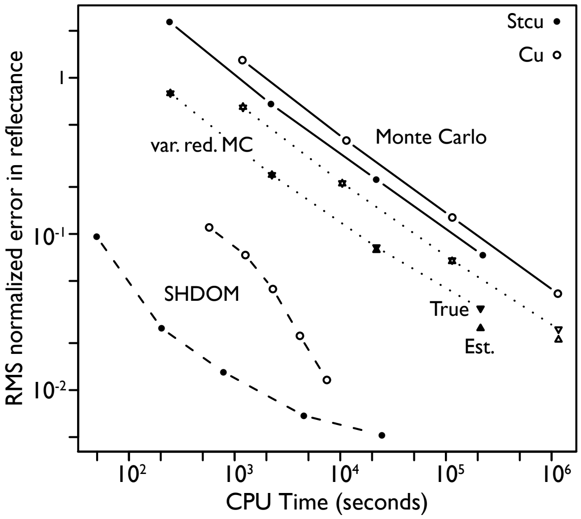

FIG. 12. Root-mean-square error in directional reflectance (normalized by the domain mean in each direction) at 0.67 μm as a function of total computation time for calculations with SHDOM (dashed lines) and the I3RC Community Monte Carlo model with (dotted lines) and without (solid lines) biased variance reduction methods. Closed symbols denote results for the stratocumulus scene; open symbols the cumulus scene. The variance reduction method decreases the error but, at the highest resolutions, the standard error (upward-pointing triangles) underestimates the true error (downward-pointing triangles). For pixel-by-pixel reflectance SHDOM is substantially more efficient than Monte Carlo.

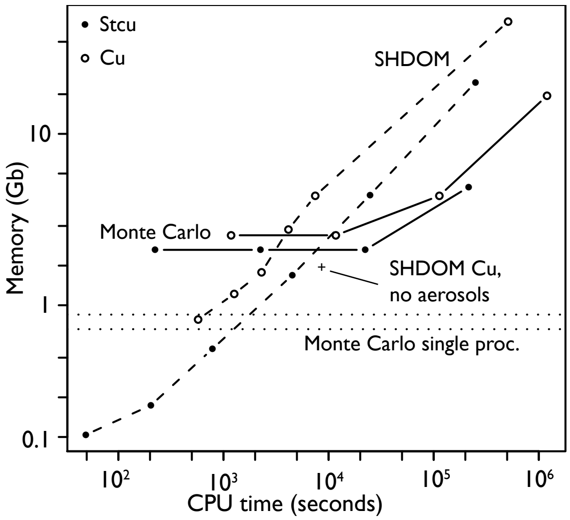

FIG. 13. Relationships between memory and computation time for the I3RC Community Monte Carlo model (solid lines) and SHDOM (dashed lineds) for the stratocumulus (closed symbols) and cumulus (open symbols) scenes. Both memory and computation time increase with accuracy for SHDOM. This need not be true for Monte Carlo calculations, and the dotted lines show the memory required for a single processor. As a practical matter, however, we increased the amount of memory by using more processors to keep the wall-clock time to solution less than 8 hours. Memory requirements for SHDOM are reduced substantially for the cumulus scene when aerosols are omitted (plus sign).